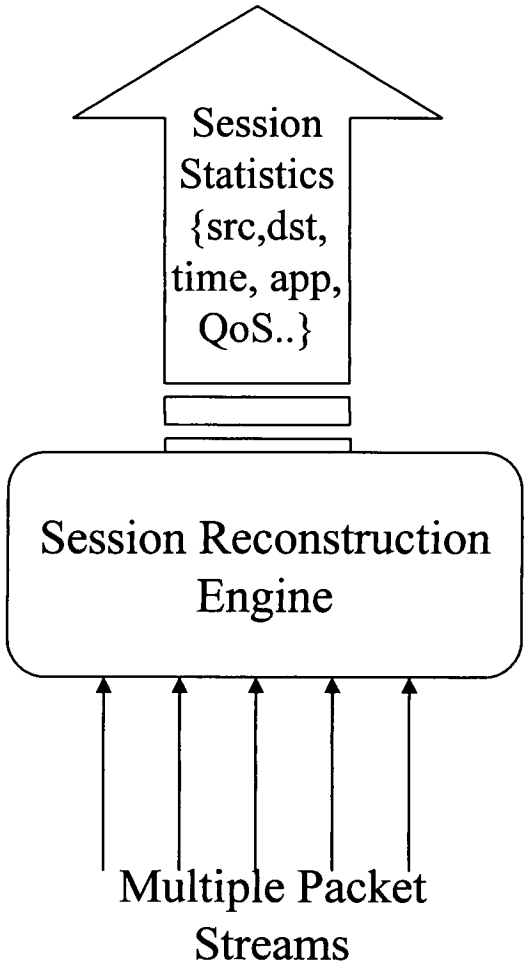
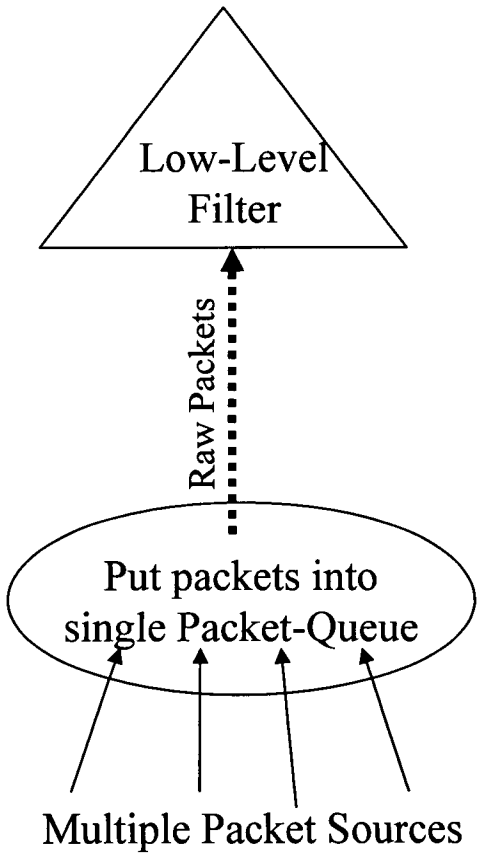
Introduction

- This is a technical paper sketching the session-reconstruction architecture.
- Unique features and highlights:
  - Application recognition based on session-streamed content.
  - Identification of session QoS metrics.
  - Multi packet-stream input
  - Architecture enables distributed collection (restrictions apply).



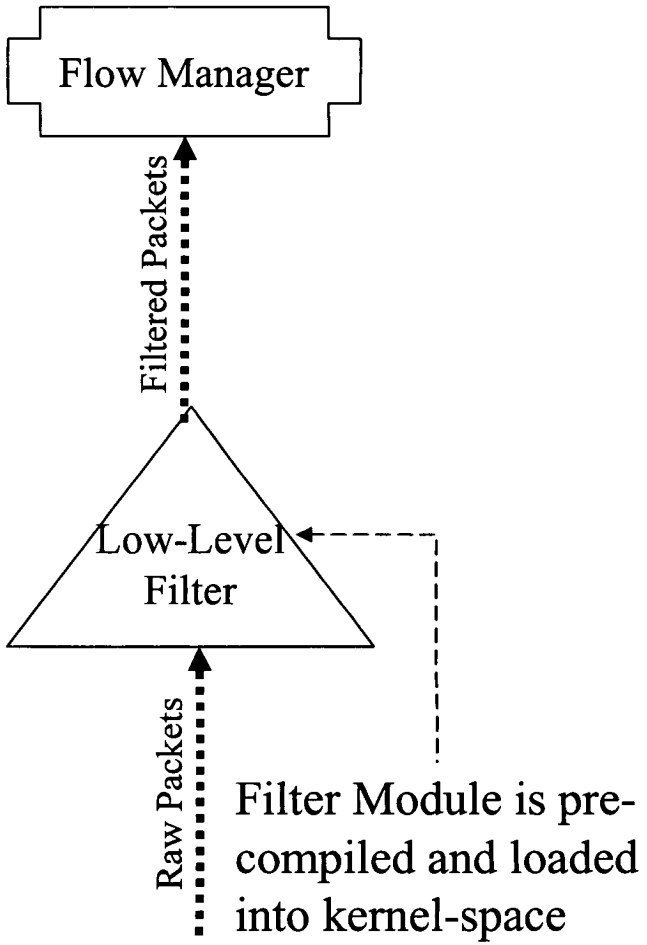
Collecting Packet Data

- Packets are collected from multiple sources (ethernet, WAN, OC192 etc.) and put into a single packet-queue
- A low-level filter is applied to the packets in order to determine if they should be neglected



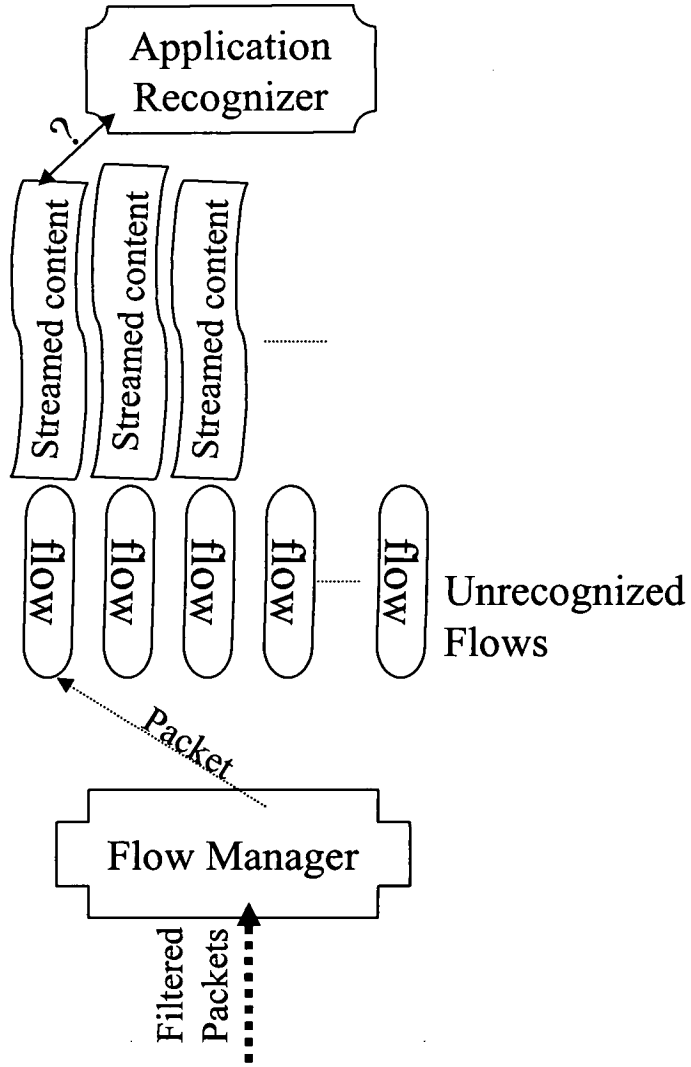
Low-Level Filter

- Low-Level Filter is intended to avoid processing "local traffic" packets
- Filter language is standard "pcap" language (ex: *tcp and port 80 or dst net 199.203.0.0 mask 255.255.0.0*) which is translated to assembly and loaded to kernel as loadable module
- Assumption: Filter module will not be created more frequently than once every "few minutes"



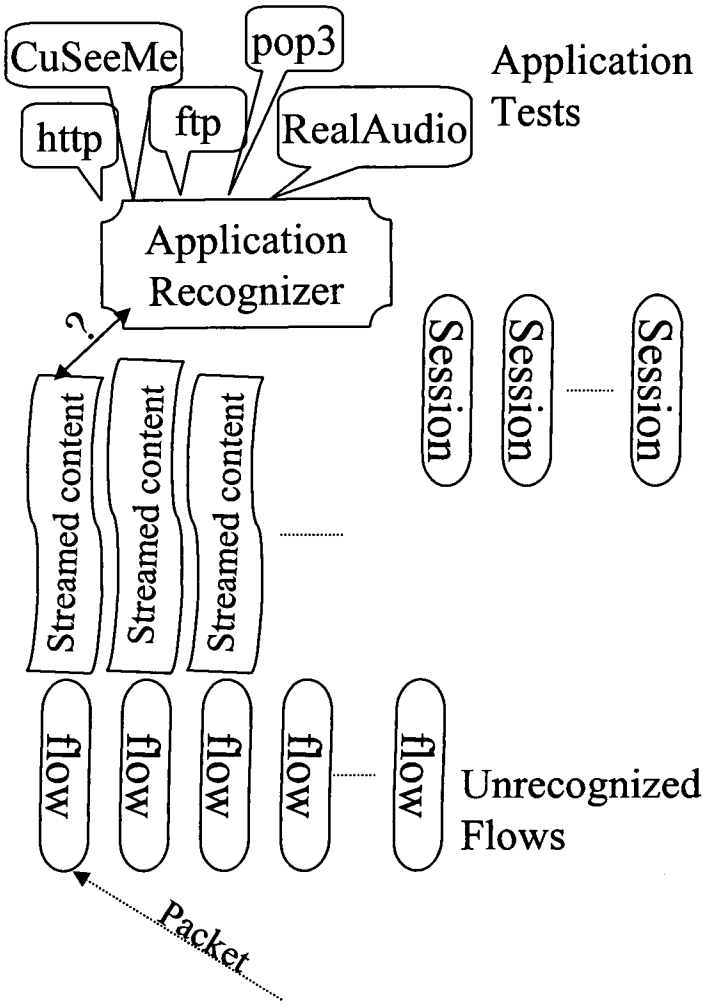
Flow Manager

- Every packet is mapped to an entry in the flow buffer based on the (maximal) key set: {src, dst, sport, dport, prot, TOS }. The key-set can be configured to any subset of the above to allow support for DIFFSERV flows or other flow definitions.
- If a flow is not already recognized or expected, the streamed data-content is collected and the Application Recognizer is queried (test performed for every packet containing data).



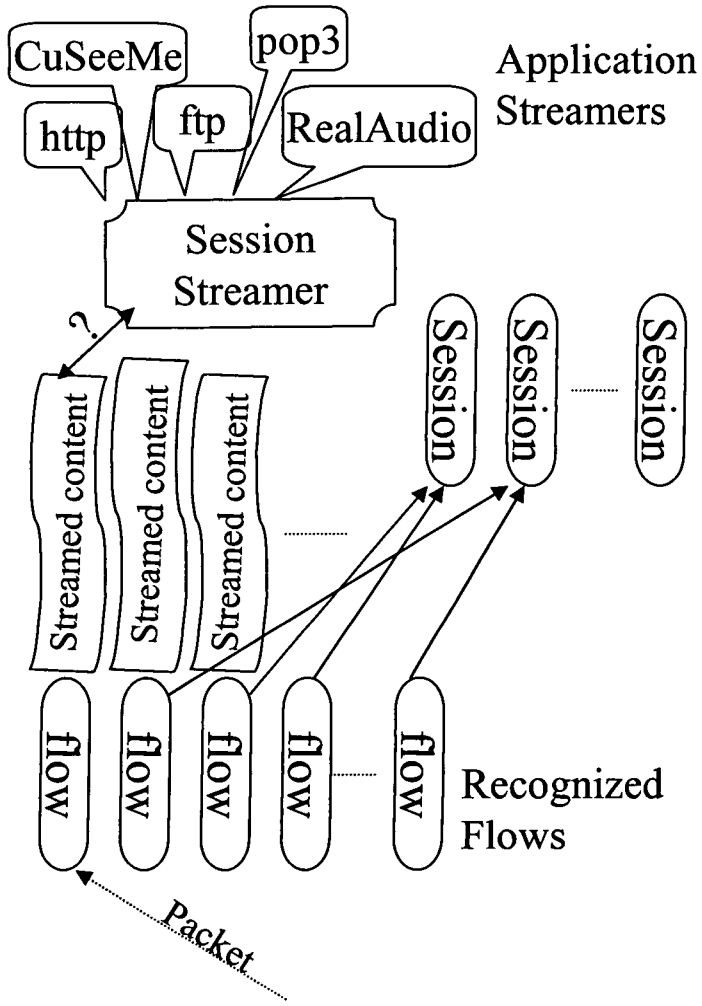
Application Recognizer

- In order to associate an application (http, RealAudio,..) to a flow, the Application Recognizer uses: up to 2k of flow streamed data, other flow-streams and flow port numbers.
- If an application is identified, a new Session is created. Else, (after 2k), the flow is taken to be a single-flow Session.
- Comment: Failed tests are not retried. Ports just help by reshuffling application-match tests.



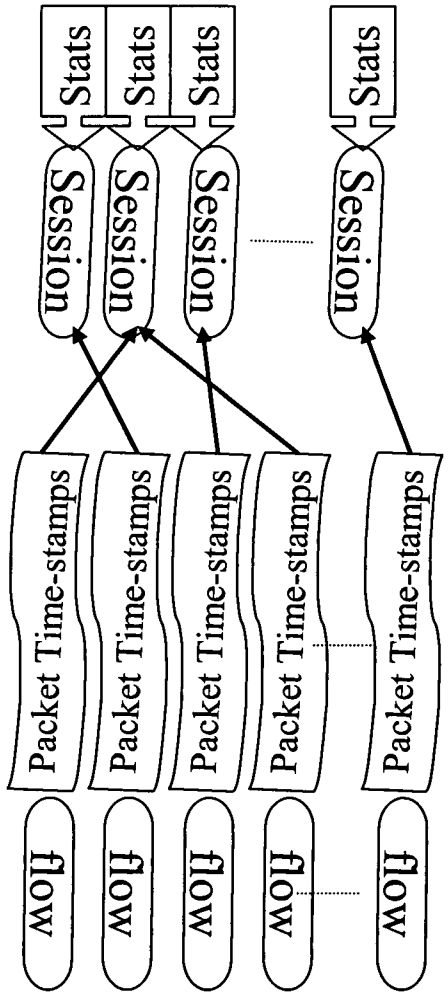
Session Streamer

- Some applications (like RealAudio) create new flows during a session. The Session-Streamer, once a session is identified, keeps track of these flows, mapping them to the right Session entry when they are formed, bypassing Application-Recognizer.
- Comment: This implies that some flows have to be data-streamed all throughout the session, while for other flows, data-streaming is not required.



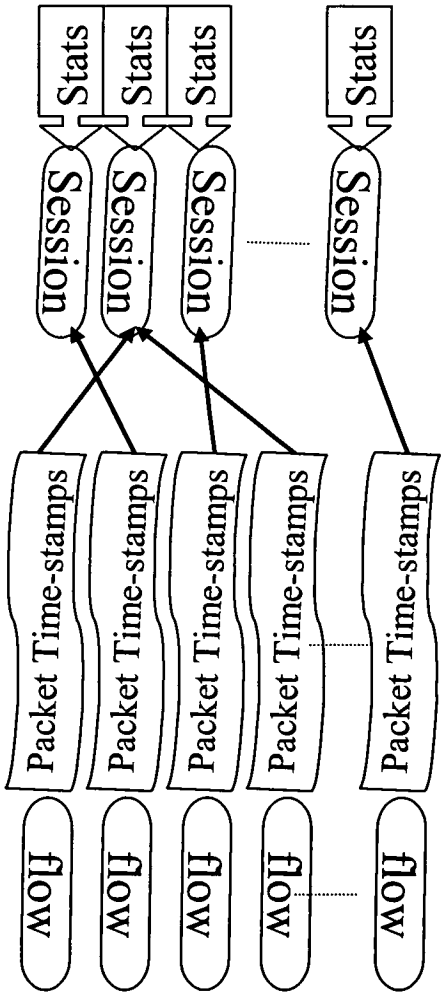
Session Statistics

- Statistics are collected for every session. Some stats are collected at flow-level, others at session (multi-flow) level, based on Packet-Timestamps that are collected with every flow.
- Collected stats (relative to both session and to last output):
  Start, # of packets, net/total bytes, avrg time between packets, moving avrg, latency (func of application), throughput, jitter (standard deviation of latency & throughput)



Latency Calculation

- Latency may be calculated differently for every application type.
- TCP latency is calculated based on sequential ACK timestamps.
- RTP flow latency is calculated based on the difference between end of communication in the control channel and beginning of communication on data channel.



Session Output

- Session-stats output verbosity is configurable. Options are: beginning of session, end of session, at application-related events and every N seconds.
- Comment: Session output granularity depends on whether needed for billing, pre-payment, data-association or real-time visualization.